I

Extraction de Chemins


SELLIER

Chapter 1

Les Chemins Minimaux en Traitement d'Images

Résumé — Les chemins minimaux sont construits à partir de l'analogie avec la propagation de la lumière dans un milieu avec un certain indice de réfraction, suivant le principe de *Pierre de Fermat*. On explique tout d'abord dans la section 1.1 le lien entre ces chemins de lumière et la théorie de la réfraction, en montrant au passage l'intéret de l'application de cette théorie au traitement d'images.
Dans la section 1.2, partant de la formulation classique des contours actifs de *Kass, Witkin, et Terzopoulos* [82], on passe à la formulation de chemin minimal, telle qu'elle a été présentée par *Cohen et Kimmel* [34].
Mettant en parallèle la formulation discrète donnée par *Dijkstra* [43] des chemins minimaux dans des graphes avec le formalisme de *Cohen et Kimmel* [34], on explicite dans la section 1.3 différentes méthodes d'extraction de chemins, ainsi que la définition des différents termes intervenant dans ce modèle: la force liée à l'image, ou *Potentiel*.
Dans la section 1.4, on étudie le role du terme qui permet de controler la longueur du chemin, et son influence sur la courbure de ce dernier.

Abstract — Minimal paths are built upon analogy with the theory of wave-light propagation in a medium with a refractive index, according to the principles of *Pierre de Fermat*. We first explain in section 1.1 the link between this minimal light paths and the refraction principle, emphasizing the interest for the use of minimal paths in image processing.
Starting from the classical formulation of the active contours called *snakes* of *Kass, Witkin, and Terzopoulos* [82], we extend in section‘1.2 to the formulation of the minimal path, as presented by *Cohen and Kimmel* [34].
Comparing the discrete version of the minimal paths given by *Dijkstra* [43] with the continuous equivalent formalism of *Cohen and Kimmel* [34], we detail in section 1.3 several implementations of extraction techniques, and the settings of parameters involved in the model, such as the image force, named *Potential*.
In section 1.4 we study the influence of the offset term which controls the length of the minimal path (and its curvature).

1.1 Minimal Paths theory

1.1.1 The minimal path in geometrical optic

In order to understand the underlying law of refraction, behind the minimal path principle, let us imagine a straight seashore, separating the sea from the beach, as shown in figure 1.1. A lifeguard sitting at a point A in the beach sees a girl drowning

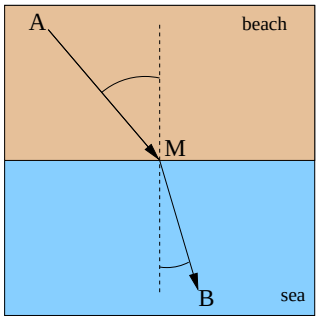


Figure 1.1. Example of minimal path in a heterogeneous media

at a point B in the sea. Assuming that the lifeguard can run on the beach three times faster than he can swim, the shortest time path is the broken line going straight from A to a point M on the shore, then straight from M to B. Considering the two angles of the straight lines to the normal to the shore, the ration of their sines is equal to the ratio of the corresponding speeds (*Descartes/Snell' law* of refraction). The principle of *Fermat* is that light waves of a given frequency travels the path between two points which takes the least time. The most obvious example of this is the passage of a light through a homogeneous medium, in which the speed of light does not change with position. In this case, shortest time is equivalent to the shortest distance between the points, which is a straight line, as shown in figure 1.2-left. When the medium is not homogeneous, as in figure 1.2-middle, there is a refraction angle at the interface between the two homogeneous regions. Figure 1.2-right can illustrate a well-known optical phenomenon, called *mirage*: The light source S is visible from both points R_1, and R_2. But the light path between S and R_2 is not a straight line, due to the difference of index of the two media. Therefore, R_2 "sees" S coming from location M, at the interface between the two media, while the image source is far from M. This phenomenon occurs when the variations in temperature are important enough to deviate the light path, resulting in "visions" of an oasis in the desert, for example. Hamilton defined optical path functions, which best known was defined by Burns as the *Eikonal equation* applied to the development of a mathematical theory of optical systems. The *Eikonal equation* is used to compute the minimal light paths for a refractive index, in the sense that the minimal path is the one which integral over the refractive index is minimal.

We are going to use this minimality property in order to extract curves in images, giving only the two extremities of the path, and using the equation developed by *Hamilton* for optical systems.

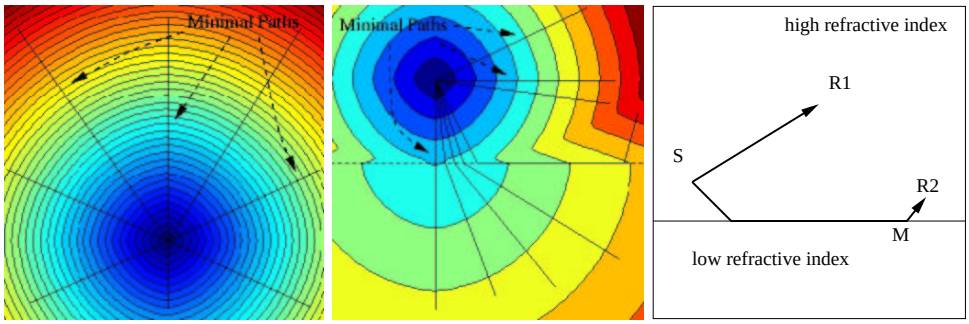


Figure 1.2. Example of minimal paths in synthetic media: left image represents the propagation of a light wave in a homogeneous medium, starting from a unique light source. The minimal paths between this light source and other position in the plane are straight lines. Middle image represents the propagation of a light wave in a medium where the refractive index is more important in the upper half part of the image than in the lower, starting from a light source in the upper part. Right image is a diagram which illustrates the *mirage* phenomenon on the basis of the propagation of the light in heterogeneous media, as shown in middle image.

1.1.2 Minimal path for curve extraction

We explain how this minimal path principle can be used for curve extraction in images. Given a refractive index P, called *potential* in the following, which takes lower values near the edges or features, our goal is to find a single contour that best fits the boundary of a given object or a line of interest. This contour, considered between two fixed extremities, will be the one which integral over the potential P is minimal.

Looking for a path which lies in a desired region of interest, the refractive index should model the desired properties of the targeted curve. For example, in figure 1.3, we want to extract a path which stays inside the vessel. The dataset, a digital sub-

Figure 1.3. Example of a minimal path in a media defined by a grey level image: The minimal path (in white) superimposed on the date is the one that corresponds to the light wave propagation, using the grey-level information as a refractive index.

tracted angiography (DSA) shows vessels in lower grey levels on a bright background. Since we want to extract a path which stays inside the vessel, a possible refractive index to be used with the *Hamilton* equations for extracting minimal paths, could be the simple image grey level values.

This 'best fit' question leads to algorithms that seek for the minimal path, i.e. paths along which the integration over P is minimal. Classical path extraction techniques are based on the *snakes* [82]. Snakes are a special case of deformable models as presented in [174]. Snakes start from a path close to the solution and converge to a local minimum of the energy. In this minimal path formulation, one interesting aspect is that the user input is limited to the end points, simplifying the initialization process. Unicity of this minimal path, for a given media avoids erroneous local minima. Motivated by the ideas put forward in [86, 87] *Cohen and Kimmel* developed an efficient and consistent method to find the path of minimal cost between two points, using the surface of minimal action [151, 87, 178] and the fact that operating on a given potential (cost) function helps in finding the solution for our path of minimal action (also known as minimal geodesic, or path of minimal potential). In the following we show how the formulation of the minimal light path can be obtained through a modification of the classical formulation of the active contours, and we show the numerical implementation of the minimal path extraction.

1.2 The Cohen-Kimmel Method in 2D

Starting from the famous *Snakes* model, we show how can we derive a formulation of a minimal path extraction which leads to the expression of one of the optical equations of *Hamilton*, the *Eikonal*.

1.2.1 Global minimum for Active Contours

We present in this section the basic ideas of the method introduced by *Cohen and Kimmel* [34] to find the global minimum of the active contour energy using minimal paths. The energy to minimize is similar to classical deformable models (see [82]) where it combines smoothing terms and image features attraction term (Potential P):

$$E(C) = \int_{\Omega} \Big\{ w_1 \|C'(s)\|^2 + w_2 \|C''(s)\|^2 + P(C(s)) \Big\} ds \tag{1.1}$$

where $C(s)$ represents a curve drawn on a 2D image, $\Omega = [0, L]$ is its domain of definition, and L is the length of the curve. Thus the curve is under the control of two kinds of forces:

- The internal forces (the first two terms) which impose the regularity on the curve. The choice of constants w_1 and w_2 determines the elasticity and rigidity of the curve.

- The image force (the potential term) pushes the curve to the significant lines which correspond to the desired attributes, for example

$$P(\mathcal{C}) = g(\|\nabla I(\mathcal{C})\|). \tag{1.2}$$

  Here, I denotes the image and $g(\cdot)$ is a decreasing function. In the classical snakes [82], we have $g(x) = -x^2$. The curve is then attracted by the local minima of the potential, i.e. edges (see [61] for a more complete discussion of the relationship between minimizing the energy and locating contours). Other forces can be added to impose constraints defined by the user. As introduced in [29], previous local edge detection [20] might be taken into account as data for defining the potential.

The approach introduced in [34] modifies this energy in order to reduce the user initialization to setting the two end points of the contour C. They introduced a model which improves energy minimization because the problem is transformed in a way to find the global minimum. It avoids the solution being sticked in local minima. Let us explain each step of this method.

1.2.2 Problem formulation

Most of the classical deformable contours have no constraint on the parameterization s, thus allowing different parameterization of the contour C to lead to different results. In [34], contrary to the classical snake model (but similarly to geodesic active contours), s represents the arc-length parameter, which means that $\|C'(s)\| = 1$, leading to a new energy form. Considering a simplified energy model without any second derivative term leads to the expression $E(C) = \int \{w\|C'\|^2 + P(C)\}ds$. Assuming that $\|C'(s)\| = 1$ leads to the formulation

$$E(C) = \int_\Omega \{w + P(C(s))\}ds \tag{1.3}$$

We now have an expression in which the internal forces are included in the external potential. In [34], the authors have related this problem with the recently introduced paradigm of the level-set formulation. In particular, its Euler equation is equivalent to the geodesic active contours [23]. The regularization of this model is now achieved by the constant $w > 0$. This term integrates as $\int_\Omega wds = w \times \text{length}(C)$ and allows us to control the smoothness of the contour (see [34] for details). We remove the second order derivatives from the snake term, leading to a potential which only depends on the external forces, and on a regularization term w. It makes thus the problem easier to solve, and it is used in minimal paths [34], active contours using level sets [113] and geodesic active contours as well [23]. In [34] is also mentioned how the curvature of the minimal path is now controlled by the weight term w. This corresponds to a first order regularization term, and the paths show sometimes angles. A second order regularization term would give nicer paths, but it is difficult to include such a term in the approach.
Given a potential $P > 0$ that takes lower values near desired features, we are looking for paths along which the integral of $\tilde{P} = P + w$ is minimal. The surface of minimal

action U is defined as the minimal energy integrated along a path between a starting point p_0 and any point p:

$$U(p) = \inf_{\mathcal{A}_{p_0,p}} E(C) = \inf_{\mathcal{A}_{p_0,p}} \left\{ \int_\Omega \tilde{P}(C(s))ds \right\} \tag{1.4}$$

where $\mathcal{A}_{p_0,p}$ is the set of all paths between p_0 and p. The minimal path between p_0 and any point p_1 in the image can be easily deduced from this action map. Assuming that potential P is always positive, the action map will have only one local minimum which is the starting point p_0, and the minimal path can be found by a simple back-propagation on the energy map. Thus, contour initialization is reduced to the selection of the two extremities of the path.

It is possible to compute the surface U in several ways. We are going to describe one of them that is consistent with the continuous case while implemented on a rectangular grid. It is, however, possible to implement a simple approximation like the shading from shape algorithm introduced in [178], or even graph search based algorithms ([43, 151]), if consistency with the continuous case is not important.

1.3 Numerical Implementation

The numerical schemes we propose are consistent with the continuous propagation rule. The consistency condition guarantees that the solution converges to the true one as the grid is refined. This is known **not** to be the case in general graph search algorithms that suffer from *digitization bias* due to the *metrication error* when implemented on a grid [119, 93]. This gives a clear advantage to our method over minimal path estimation using graph search. Before we introduce the proposed method, let us review the graph search based methods that try to minimize the energy given in (1.3).

1.3.1 Graph Search Algorithms and Metrication Error

Based on the new energy definition (1.3), we are able to compute the final path without evolving an initial contour, by using the surface of minimal action. To find the surface of minimal action, graph search and dynamic programming techniques were often used, where the image pixels serve as vertices in a graph [122, 53, 26], considering the image as an oriented graph.

Oriented graph

A digital image is an array of pixels. With the optimal path approach, the pixel-array is considered to be a directed graph. A local cost is associated with each node of the graph, and a link cost is associated with every arc. The costs (both local and link) are determined by the energy to minimize, therefore also called cost-function. The problem of finding the optimal boundary segment between two image pixels is transformed into finding the optimal path between two nodes in the graph. A path between two points is said optimal if the sum of the cost function values at

each of its points (called cumulative cost) is lower than the cumulative costs of any other path between the same two points. Number of algorithms, based on dynamic programming, are available in the graph theory literature. Dijkstra's algorithm [43], which computes the optimal path between a single source point to any other point in the graph, is the basis of both the *Live-Wire* and the *Intelligent Scissors* 's graph search algorithms [49, 127].

Dijkstra's path search algorithm

Path extraction algorithms like *Live-Wire* [49] and *Intelligent Scissors* [127] tools use a search method based on Dijkstra's algorithm [43]. A description of Dijkstra's algorithm, applied to road detection, can be found in [53].

The algorithm (see table 1.1) is initialized by selecting a start point s (also called seed point) in the graph. Giving this start point s, the cumulative cost of a pixel p is the sum of the cost function values of the points of the optimal path from s to p. The size of the active list is the total range of possible cumulative cost values, and its i-st item contains the list of pixels with their cumulative cost valuing i. The cumulative cost is initialized with value ∞ everywhere except at a start point s with value zero. The active list is initialized by inserting the starting pixel into the first sub-list.

At each iteration of the algorithm, the pixel p with the lowest cumulative cost is removed from the active list and expanded: the cumulative cost of each of its non-expanded neighbors is updated, and the active list is reorganized. The updating of the cumulative cost of the neighbors runs as follows: the newly computed cumulative cost of a neighbor q is the sum of the cumulative cost of its father p and the link cost from p to q. If this newly computed cost is lower than the previous one the cumulative cost of q becomes the lately calculated cost, the active list is updated and a path map structure keeps a pointer from q back to p. Since at each iteration one pixel gets a final value, and a search for the minimal vertex to update is performed, the algorithm complexity is $O(N \log_2 N)$ where N is the number of pixels in the image. In addition, a marker registers all the expanded pixels. The optimal path between the start point and another point in the image is obtained by following the pointers of the path map from the end point back to the seed point.

Our approach solves a continuous version of the problem. Sethian Fast Marching Method [161], described in section 1.3.2, has a similar complexity, yet it is consistent!

1.3.2 Fast Marching Resolution

In order to compute this map U, a front-propagation equation related to equation (1.4) is solved :

$$\frac{\partial C}{\partial t} = \frac{1}{\tilde{P}} \overrightarrow{n} \tag{1.5}$$

It evolves a front starting from an infinitesimal circle shape around p_0 until each point inside the image domain is assigned a value for U. The value of $U(p)$ is the time t at which the front passes over the point p.

Algorithm for optimal path search

- Input:
  - cost function $\mathcal{P}$ ($\mathcal{P}(p,q)$ = cost of the arc (p,q) in the graph);
  - starting seed pixel s;
- Output: the path map $\mathcal{M}$;
- Data structures:
  - cumulative cost array CC
  - Sorted active list $\mathcal{L}$
  - Boolean expanded pixels marker $\mathcal{E}$
- Begin:
  1. put $CC(s) = 0$ and $CC(n) = \infty\ \forall n$ in the set of graph vertices;
  2. put s in the list $\mathcal{L}$;
  3. while $\mathcal{L}$ is not empty do
     (a) consider p the pixel in $\mathcal{L}$ with $CC(p)$ minimal;
     (b) consider $\mathcal{E}p$ = TRUE;
     (c) remove p from $\mathcal{L}$;
     (d) For each neighbor q of p such that $\mathcal{E}p$ = FALSE, do
        i. $u = CC(p) + \mathcal{P}(p,q)$;
        ii. if $u < CC(q)$ then
           - $CC(q) = u$;
           - update position of q in $\mathcal{L}$;
           - put a marker from q to p in $\mathcal{M}$.

Table 1.1. Dijkstra Optimal Path Search Algorithm as used in [49] and [127]

The *Fast Marching* technique, introduced in [2], [161], and detailed in [163], was used by [33], noticing that the map U satisfies the Eikonal equation:

$$\|\nabla U\| = \tilde{P} \tag{1.6}$$

originally developed by *Hamilton and Burns* for geometrical optics (see section 1.1). Classic finite difference schemes for this equation tend to overshoot and are unstable. [163] has proposed a method which relies on a one-sided derivative that looks in the up-wind direction of the moving front, and thereby avoids the over-shooting associated with finite differences :

$$\begin{aligned} (\max\{u - U_{i-1,j}, u - U_{i+1,j}, 0\})^2 \quad &+ \\ (\max\{u - U_{i,j-1}, u - U_{i,j+1}, 0\})^2 \quad &= \quad \tilde{P}_{i,j}^2 \end{aligned} \tag{1.7}$$

giving the correct viscosity-solution u for $U_{i,j}$. Authors of [150] also presented a direct numerical approach to solve (1.6) and gave a convergence proof to that minimization procedure in the viscosity solutions framework [35]. The principle of the *Fast Marching* is to introduce order in the selection of the grid points. This order is based on the

fact that information is propagating *outward*, because action can only grow due to the quadratic equation (1.7). Therefore the solution of equation (1.7) depends only on neighbors which have smaller values than u.
The algorithm is detailed in 3D in next section in table 2.1. The *Fast Marching* technique selects at each iteration the *Trial* point with minimum action value. This technique of considering at each step only the necessary set of grid points was originally introduced for the construction of minimum length paths in a graph between two given nodes in [43].
Thus it needs only one pass over the image. To perform efficiently these operations in minimum time, the *Trial* points are stored in a min-heap data-structure (see [161, 160, 1, 89, 163] for further details on the above algorithm, as well as a proof of correct construction). Since the complexity of the operation of changing the value of one element of the heap is bounded by a worst-case bottom-to-top proceeding of the tree in $O(\log_2 N)$, the total work is about $O(N \log_2 N)$ for the *Fast Marching* on a N points grid.
Finding the shortest path between any point p and the starting point p_0 is then simply done by back-propagation on the computed minimal action map. It consists in gradient descent on U starting from p until p_0 is reached, p_0 being its global minimum, since the geodesics are orthogonal to the wave fronts (see Bellman [11] for a nice proof on this orthogonality).

1.3.3 Back-propagation

In order to determine the minimal path between p_0 and p_1, we need only to calculate U_0 and then slide back on the surface U_0 from $(p_1, U_0(p_1))$ to $(p_0, 0)$. The surface of minimal action U_0 has a convex like behavior in the sense that starting from any point $(q, U_0(q))$ on the surface, and following the gradient descent direction, we will always converge to p_0. It means that U_0 has only one local minimum that is of course the global minimum and is reached at p_0 with value zero.

This is a consequence of the results in [11] that show that the light rays (geodesics, constant parameter curves) are orthogonal to the wave fronts (equal cost contours). The gradient of U is therefore orthogonal to the wave fronts since these are its level sets. The back propagation procedure is a simple steepest gradient descent. It is possible to make a simple implementation on a rectangular grid: given a point $q = (i, j)$, the next point in the chain connecting q to p is selected to be the grid neighbor (k, l) for which $U(k, l)$ is the minimal, and so forth.

1.3.4 Comparing Dijkstra and Eikonal

Since there is no difference in the overal complexity of both implementations, one may argue that using previously mentioned graph search algorithm like Dijkstra's [43, 155] might be sufficient. This algorithm is indeed efficient, but suffers from metrication errors. The graph based algorithms consider the image as an oriented graph in which each pixel is a node, and the 4 (or 8) connections to the neighboring pixels are the vertices of the graph. The different metrics used lead to very different results in a homogeneous media (see figure 1.4). Because in the L_1 metric considered for Dijkstra,

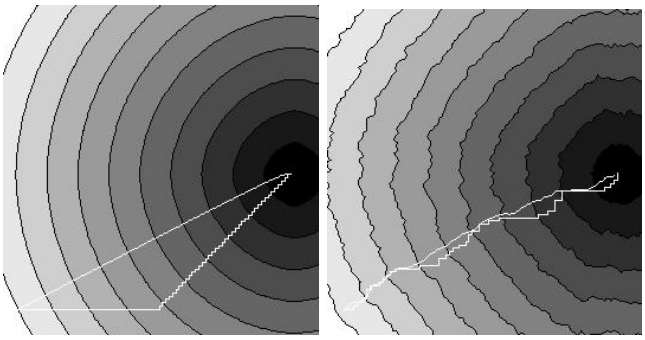


Figure 1.4. Comparing Dijkstra and Eikonal optimal path extraction in a homogeneous media Left image represents both paths with the iso-action levels. Right image is the same with an added Gaussian noise.

we are limited by the distance measure imposed by the graph, how fine the grid gets. With the 'right' Euclidean distance measure (L^2) we get the diagonal connection as the optimal path in this case. However, notice that the homogeneous media is a synthetic dataset, and that in a noisy image, both discrete and continuous formulation lead to similar results (see figure 1.4-right).

Of course the result of the graph-search could be improved by taking a larger neighborhood as structuring element [16, 176], but there will always be an error in some direction that will be invariant to the grid resolution, which is not the case in the discretization of the *Eikonal* equation.

This error is still too important for reasons of accuracy in medical applications for example. As an illustration, figure 1.5 shows the result of searching for the minimal path in a test image.

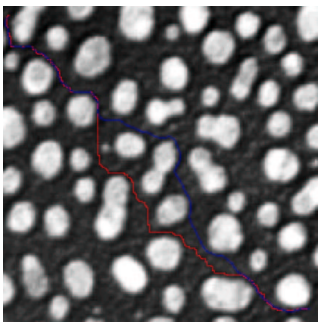


Figure 1.5. Difference between Dijkstra and Eikonal on a real image: The difference is obvious in this real image example. The potential used for computing the action is the grey level information. Using the same extremities, the red path corresponds to Dijkstra's, and the blue one to Eikonal.

1.4 The Regularity of the Path

In [34], it is proven that weight w in equation (1.3) can influence curvature and be used as a smoothing term. An upper bound for the curvature magnitude $|\kappa|$ along the minimal path is found, $\mathcal{I}$ being the image domain:

$$|\kappa| \leq \frac{\sup_{\mathcal{I}} \|\nabla P\|}{w} \tag{1.8}$$

1.4.1 Influence on the gradient descent scheme

The exact minimal path is obtained with a gradient descent. But care must be paid on the choice of the gradient step to avoid oscillations.

If the weight w is set to a small value ϵ the extracted path length is not limited at all, nor the curvature magnitude in equation (1.8). Therefore in zones where the action map is flattened, the slope being as small as ϵ, the path can have a spaggethi-like trajectory. The minimal path being obtained by steepest gradient descent, directions are evaluated by interpolation based on nearest neighbors on the Cartesian grid. If the discrete gradient step $\Delta\mathbf{x}$ is too large, the approximation of this trajectory will produce oscillations between relative positions. Those oscillations can lead to a huge number of path points larger than forecasted allocations.

We have made a test on a region of the data shown in figure 1.6-left where the steepest gradient fails (with a number of path points limited). The cost map when

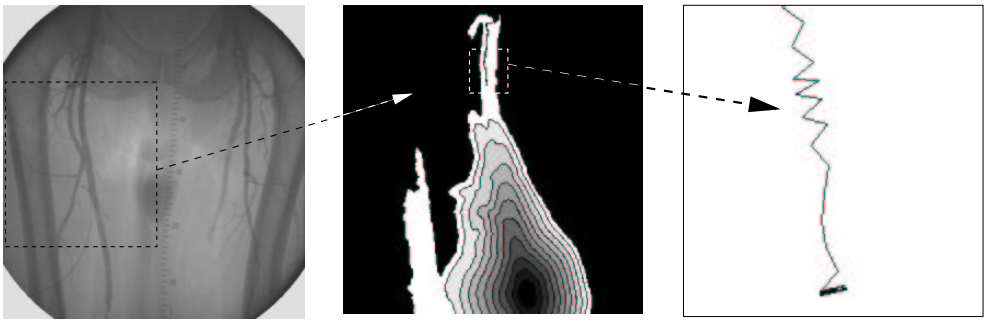


Figure 1.6. Failure of the steepest gradient descent on a bolus chase reconstruction data

tracking a vessel is displayed in figure 1.6-middle. Taking $w = .1$ leads to a curvature magnitude $\kappa \leq 10^3$. The steepest gradient scheme oscillates, for a given step size, and stops as shown in figure 1.6-right. Therefore, increasing w maintains a lower upper-bound on the curvature magnitude and makes the steepest gradient descent scheme robust. Another method is to use more robust gradient descent techniques like Runge-Kutta where the step size of the gradient descent can be locally adapted.

1.4.2 Influence on the number of points visited

This section illustrates the influence of the weight w of equation (1.3) on the necessary number of voxels visited for a path extraction. In figures 1.7 is shown the tracking of a vessel in a X-Ray image of the femoral vessels, using different weights $w_1 = 1$ and $w_2 = 20$. The smoothing done by increasing the weight can be observed in a

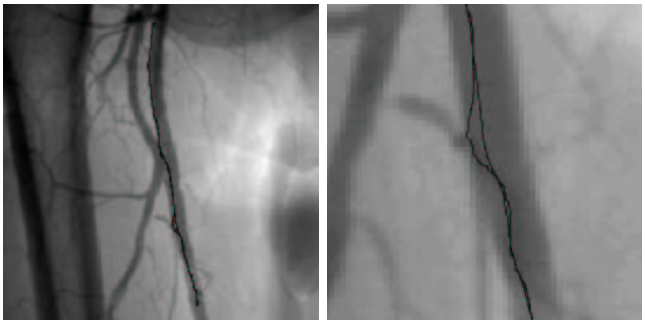
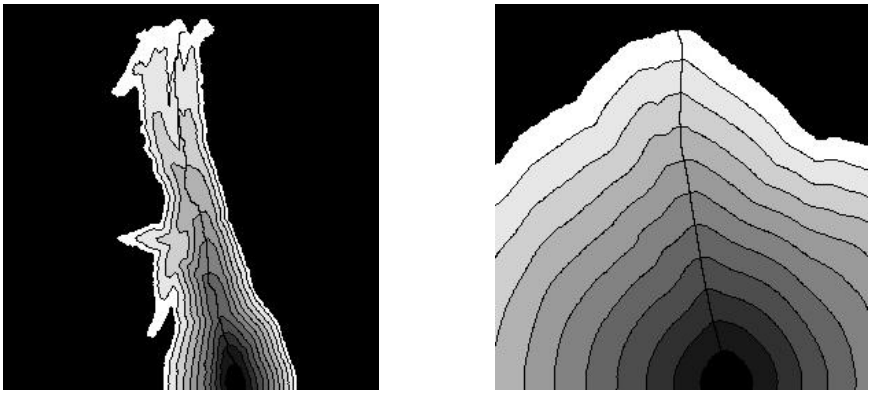


Figure 1.7. Smoothing the minimal path with the weight w: Left image shows the dataset of femoral vessels with two paths superimposed. Right image shows a zoom on the paths with $w_1 = 1$ and $w_2 = 20$.

zoom on the paths shown in figure 1.7-right. We can also observe the influence of increasing the weight in figure 1.8 where each path is displayed superimposed on its respective action map. For a small weight $w_1 = 1$, the path is not smoothed, as shown



$w_1 = 1$, and $|\kappa_1| \leq 15.4$ $w_2 = 20$, and $|\kappa_2| \leq 0.75$

Figure 1.8. Smoothing the minimal path with the weight w: the action maps

in figure 1.8-left. For a weight $w_2 = 20$, leading to the inequality $|\kappa_2| \leq 0.75$, the path is smooth. Differences appear also in the sets of points visited during propagations: it is smaller with weight $w_1 = 1$. It means that propagation is quicker for small weights. It propagates in every directions for a higher weight (see figure 1.8-right), because the

tune of w smoothes the image, as it reduces the upper-bound on curvature magnitude in equation (1.8).